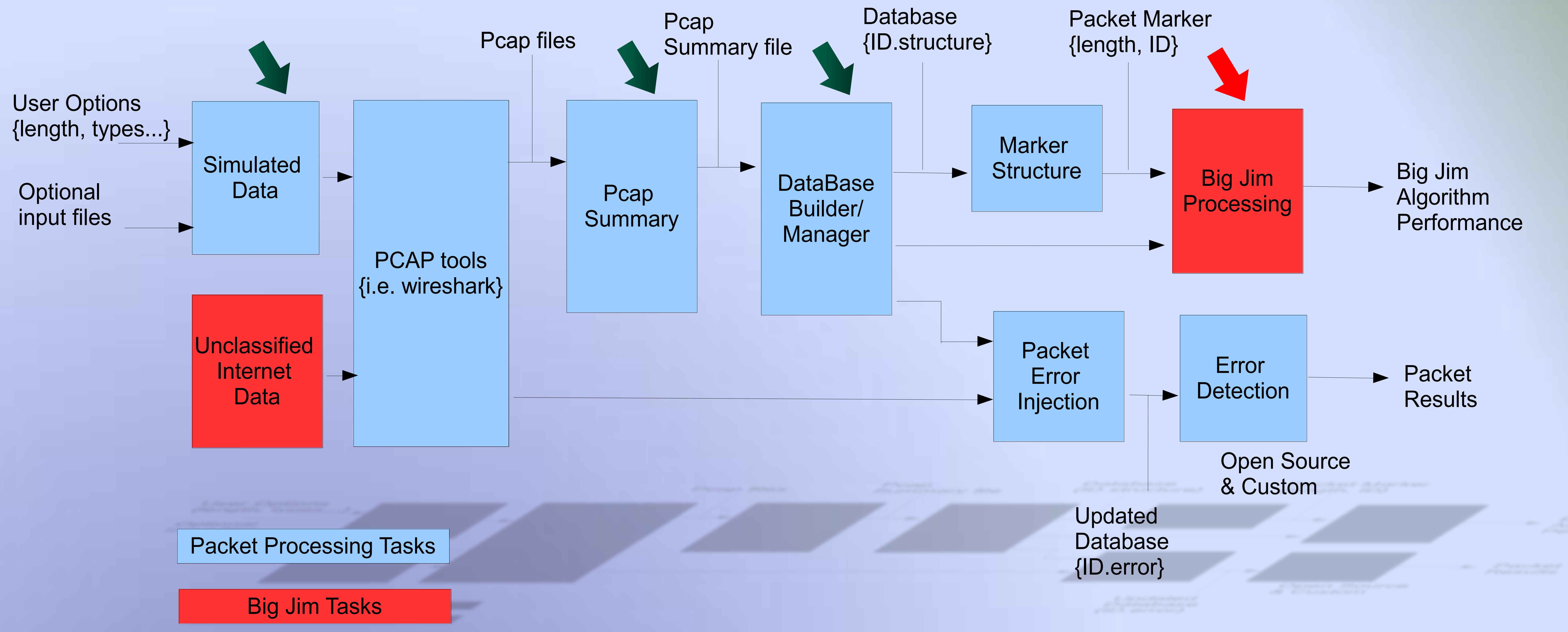
# Packet Error Injection Test-Bed

**Lawrence Livermore National Laboratory**
Alan Yang, Cal Poly San Luis Obispo
Chris Gardner, Lawrence Livermore National Laboratory



User Options {length, types...}

Optional input files

Simulated Data

Unclassified Internet Data

PCAP tools {i.e. wireshark}

Pcap files

Pcap Summary file

Pcap Summary

Database {ID.structure}

DataBase Builder/ Manager

Marker Structure

Packet Marker {length, ID}

Big Jim Processing

Big Jim Algorithm Performance

Packet Error Injection

Error Detection

Packet Results

Open Source & Custom

Updated Database {ID.error}

Packet Processing Tasks

Big Jim Tasks

## Introduction

My goal is to develop a test-bed consisting of tools for backend packet processing as well as a framework to allow for packet error injection. Together, these tools will generate and quantify traffic, identify packet errors, and most importantly of all, provide useful metrics.

Ultimately, this set of tools will be compared against the "Big Jim" project as a test of Big Jim's effectiveness over "traditional" packet processing techniques.

## Methods

Tools:

To create these tools, I relied heavily on adapting open-source software to my needs. My goal was not to recreate the wheel but to quickly develop useable tools. Where functionality was lacking, I added it by writing wrapper scripts or editing the open source code directly.

**Packet Injector:**
This tool currently allows for injection of packets of 6 different kinds of protocols. This tool makes use of the open-source tool Nemesis with a few adjustments to its code as well as added functionality for sending large payload files in chunks.

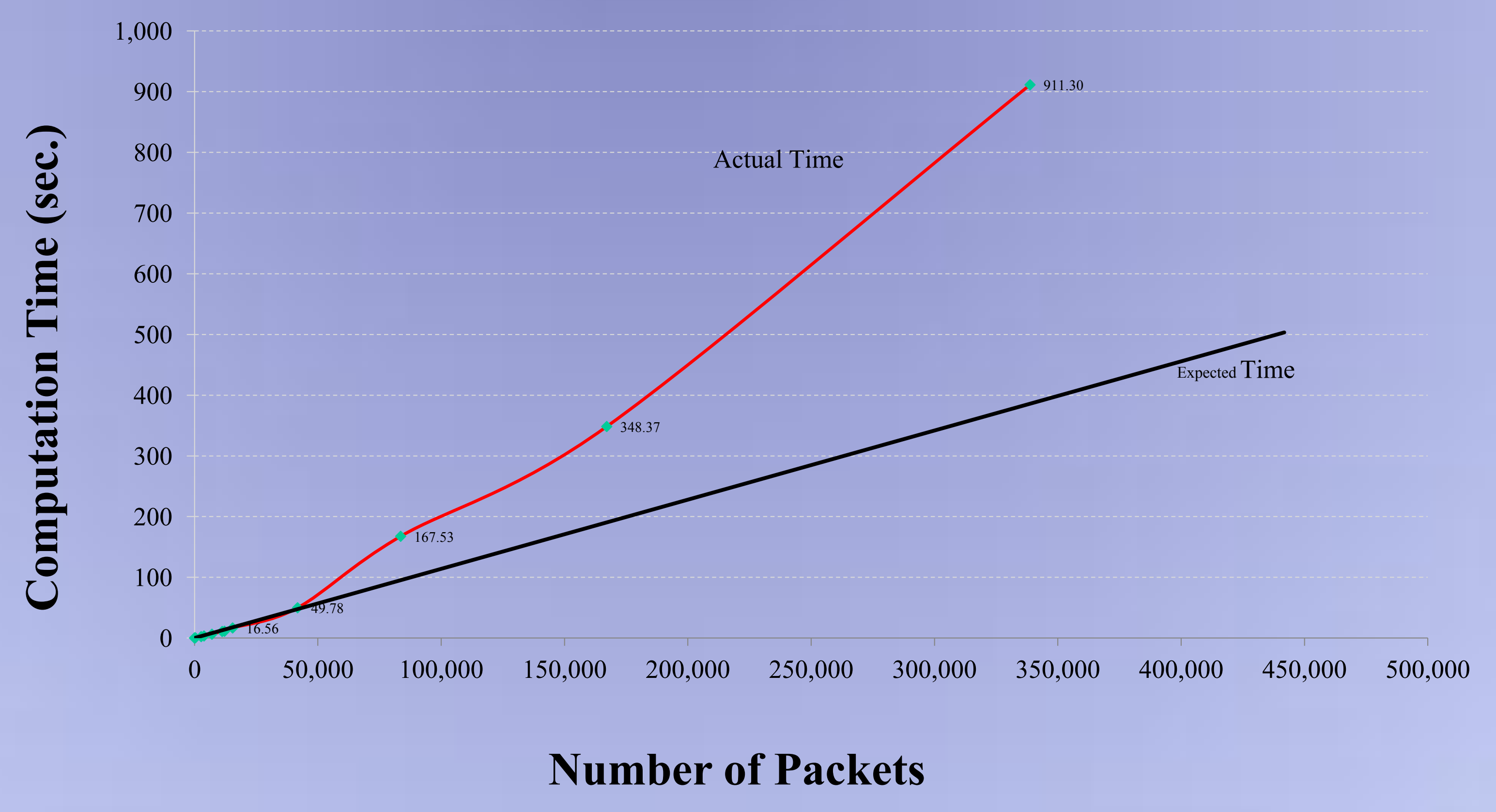
**Pcap Summary Tool:**
Wireshark performs an admirable job of sniffing out packets going to and from the computer. The summary tool takes the .pcap file resulting from a Wireshark capture and gives the user the ability to parse traffic data based on the characteristics he/she is looking for: i.e. protocol. The result is an easy-to-parse CSV file ready for insertion into a database.

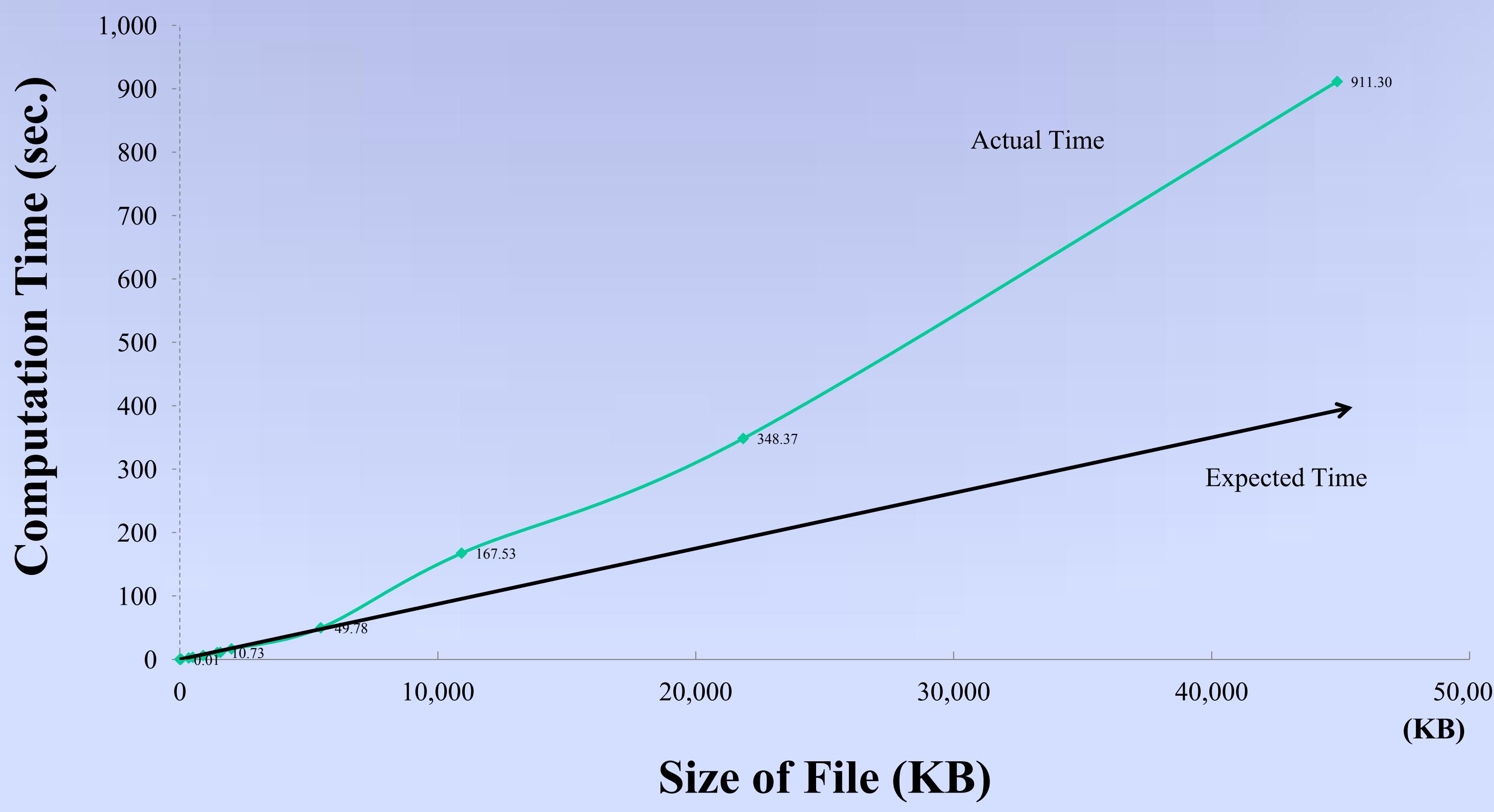
**Packet Database Tool:**
This tool builds a database (files) of packet data based on particular characteristics of interest. Each unique subset of packet characteristics will have its own file in the Database, summarizing all the packets that have been analyzed using the particular subset of characteristics.

## Results

Metrics of Interest:

Currently, my Metrics of Interest reflects the speed of my backend processing tools





## Discussion

So far, I am able to inject normal packets, process them and store them in a database. This is the foundation needed before implementing a packet error injection system.

Future Work: Looking at Packet Error Injection allowing for classification analysis.

Hypothesis: "Future work will show that error injections can be seen with traditional packet processing but malware injection would require more advanced packet analysis."

**Malware Emulation via Error Injection**
The meaning of "Error Injections" is a packet that is malformed. A malformed packet is easy to detect due to internal controls to ensure packet integrity such as checksums. A packet that is maliciously altered will undoubtedly fix the checksums so as to evade error detection.

**Big Jim Processing**
So what is the point of error injecting? In support of the "Big Jim" project, it is beneficial to show that there is a need for the functionality that Big Jim provides as well as providing a database built by "traditional packet processing methods" for Big Jim to measure against.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. DE-AC52-07NA27344

LLNL-POST-571255